“Fishbone”

Fishbone
Technical Design Document

version 1.2

January 15, 1997

Stormfront Studios, Inc.
4040 Civic Center Dr., Suite 300
San Rafael, CA, 94903
Phone (415) 479-2800
Fax (415) 479-2880

Technical Development Staff

Primary Contact

Mark A. Fong

Phone (415) 479-2800 x 216

Fax (415) 479-2880

Programmers

Mark A. Fong

Full time

4 years experience

Racing games, 3-D programming.

Andretti Racing (PSX), Mario Andretti Racing (SEGA), Eagle Eye Mysteries - The Original (MAC) Operation Tsunami (MAC).

Michael Chow

Full time

B.A. University of California at Berkeley, Cognitive Science

Additional Programmer TBD based on precise project timing. Will be full-time assignment for several months of project.

Designer

Electronic Arts & Stormfront Studios

Artists

Art Director -- Sergio Lobato. 6 years game industry experience. Gateway To The Savage Frontier (PC), Treasures Of The Savage Frontier (PC), Eagle Eye Mysteries The Original (PC, PC CD-ROM, MAC, and MAC CD-ROM), Eagle Eye Mysteries in (PC, PC CD-ROM, MAC, and MAC CD-ROM), Tony LaRussa Baseball 3 (PC CD-ROM), Andretti Racing (PSX).

Art Director -- Jim Larsen. 9 years game industry experience. Andretti Racing (PSX), Mario Andretti Racing (SEGA), Tony LaRussa Baseball 3 (PC CD-ROM), Old Time Baseball (PC CD-ROM), ESPN Hockey Tonight (PC CD-ROM), and Deep Space Nine: Harbinger (PC CD-ROM).

Artists (cont)

3D Artist -- John Chiu. 20 years art experience. Andretti Racing (PSX).

3D Artist -- Michael Cincotta. 2 years game industry experience. Madden 97 (PC CD-ROM)

3D Artist -- Michael Barber. 2 years game industry experience. Tony LaRussa Baseball 3 (PC CD-ROM), ESPN Baseball (PC CD-ROM), Old Time Baseball (PC CD-ROM), and Deep Space Nine: Harbinger (PC CD-ROM).

Billy Sullivan. 5 years game industry experience. Tony LaRussa Baseball '95 (SEGA), Tony LaRussa Baseball 3 (PC CD-ROM), Eagle Eye Mysteries The Original (PC, PC CD-ROM, MAC, and MAC CD-ROM).

Hugo Kobayashi. 3 years game industry experience. Tony LaRussa Baseball 3 (PC CD-ROM), ESPN Baseball (PC CD-ROM), and Deep Space Nine: Harbinger (PC CD-ROM).

Olga Chudnovsky. 1-1/2 years game industry experience. Old Time Baseball (PC CD-ROM), and Deep Space Nine: Harbinger (PC CD-ROM).

Individual Artists TBD

Video Production

TBD

Audio

TBD

Data Producer

TBD

Technical Producer

Ken Zarifes

Producer

Dave Ross

Target Machine

Sony PlayStation (PSX) 32-bit 33mHz MIPS R3000 CPU, 2MB RAM, 1 MB VRAM, 500K Sound RAM.

Shipping Media

To be shipped on one (1) 625 megabyte, encrypted CD-ROM.

Development Environment

Hardware

Sony PlayStation Target Box DTL-H505

CPU :	Gateway 2000 P5-120 - 120mHz Pentium CPU, 32 Megabytes of RAM, 1 Gigabyte SCSI Hard Drive, CD-ROM
Monitor:	Gateway CrystalScan 1776 LE - 17” SVGA Monitor
TV:	20” Sony Trinitron.
Cards:	DTL-H201 - Graphics Artist Board
	Sony Target Box Interface board
CPU:	Apple Power Macintosh 8100/100, 72 Megabytes of RAM, 700 Megabyte Hard Drive
Monitor:	Radius IntelliColor Display/20e
Cards:	DTL-H700 Sound Artist Board

Software

Compiler:	GNU C Compiler version 1.02 (PC)
	Symantec C Compiler version 8.01 (MAC)
Art:	Adobe Photoshop version 3.0 (MAC & PC)
	Custom 3D Track creation tool (MAC)
	Custom 3D Modeler (MAC)

Performance

Average frame rate - main game	20 frames per second
Minimum frame rate - main game	15 frames per second
Average frame rate - video scenes	15 frames per second (minimum)

Memory Map Requirements

Action Game:

- Main Memory (2MB):
  - 300K executable code
  - 15K stack space
  - 88K miscellaneous heap space
  - 32K 3-D engine data tables
  - 1000K track data 3-D points, face information, camera information and tables.
  - 200K PSX 3D data structures
  - 15K driver data (for all 24 cars)
  - 200K instant replay buffer (20 seconds of IR)
- Video RAM (1 MB):
  - 300K frame buffer (320 x 240 16-bit double-buffered)
  - 10K color look-up tables (CLUT)
  - 690K texture pages
- Sound RAM (500K)
  - 512K sampled sounds

Video Sequences:

- Main Memory (2MB):
  - 50K executable code
  - 15K stack space
  - 85K miscellaneous heap space
  - 1500K CD-ROM file buffer for streaming data (video, sound, and music) off CD.
  - 350K unallocated - for any additional features
- Video RAM (1 MB):
  - 300K frame buffer (320 x 240 16-bit double-buffered)
  - 700K unallocated - for any additional features

Setup Screens:

- Main Memory (2MB):
  - 300K executable code
  - 15K stack space
  - 85K miscellaneous heap space

- 600K CD-ROM file buffer for streaming data (screens, video, and music) off CD.
- 1000K image data (drivers portraits, pop-up screens, etc...)

Video RAM (1 MB):

- 300K frame buffer (320 x 240 16-bit double -buffered)
- 10K color look-up tables (CLUT)
- 690K texture pages

Sound RAM (500K)

- 100K MIDI music (w/ instruments)
- 100K sampled sounds
- 300K announcer samples

Configuration

Two (2) 14-button controllers. NTSC supported. Save game card to be supported.

Optional third-party analog steering controller supported also.

Compression Technology

Video will use the PlayStation's JPEG hardware compression technology to decompress individual frames to VRAM . Expected compression rate: 85% savings.

Localization Considerations

To localize this product to foreign markets, extensive voice-overs must done for the 50+ minutes of video planned. Only the text on setup screens needs to be modified. Fonts will mostly be used in the action game to display position, gear, rpm, lap times, etc... and should not need to be modified.

Software Theft Protection Plan

Encrypted CD-ROM (need I say more!)

Current Progress

Andretti Racing PSX action game will be used as the core of the product. Technical hurdles on Menu side has already been jumped in Andretti Racing PSX.

Code Source Archival

GNU C Compiler

PSX Tools (Art and sound)

Adobe Photoshop

Terrain Editor (written by Stormfront Studios)

3-D Modeler (written by Stormfront Studios)

Derivative Work Portability Considerations

Spin-off Titles

3-D Rendering engine, track rendering code, and object rendering code is written to be reusable and customizable.

Ports

3-D Rendering engine, track rendering code, and object rendering code, car physics, and driver artificial intelligence will be written in C and machine independent. Track geometry, artwork, and video will all be easily transferable to other platforms.

Porting this product will only require rewriting the most basic of drawing and fixed point math routines and some re-compilation of data tables if the platform is not in little-endian format. Modification of the geometry and artwork may be necessary if the target platform cannot match the horsepower of the PSX.

Resource ROM Budget (625 Mbytes)

Video (with sound) ~22+ minutes	440MB
CD soundtrack (redbook audio) ~10+ minutes	100MB
Track Data	16MB
Textures/Background Graphics	12MB
Sound Effects/Music	4MB
Code	1MB
Setup Screens/Misc data	10MB
Dummy Data (3 minutes)	30MB
Total	**613MB**

Major Product Milestones

Milestone #1 Delivery

December 20, 1996

Description

Sample NASCAR Car on Andretti Track - Updated 3-D stock car model with new rendering effects for an improved look.

Major Sub-modules

Build new 3-D NASCAR model
Add Gouraud shading
Add lighting effects
Animate roof flaps

Implementation Strategy

This milestone will be achieved by using the 3-D rendering engine from Andretti Racing PSX and adding new effects and an enhanced 3-D model. Gouraud shading is a built-in feature on the Sony PlayStation hardware; NASCAR PSX will now use this feature in conjunction with standard light-sourcing algorithms. Animating the roof flaps is simply an extension of the independent object rotation feature used to rotate and steer the tires on the cars.

Risks and Assumptions

Gouraud shading requires slightly more RAM and takes more time to render. Also the light-sourcing algorithms will add to the CPU load.

Dependencies

Getting accurate NASCAR blueprints.

Test for Milestone Completion

Texture-mapped car can move independently of the camera view and all objects on the car will draw in the proper order (e.g. rear tires shouldn't draw over front tires, etc...). Moving the car should produce different shading effects. Tires should spin and front tires should rotate to emulate steering. Also, the roof flaps should move independently although no physics will be attached to it until a later milestone.

Game Features Included

Accurate NASCAR model
Gouraud shading and Light Sourcing
Roof Flaps

Major Product Milestones (cont)

Milestone #2 Delivery
January 27, 1997

Description
Track Rendering Upgrade - This upgrade will try to achieve three things: improve the track rendering effect, take less CPU time to do it allowing for more cars to be on the track, and lessen the RAM usage to allow for larger or more complex tracks.
Skid Marks - Code to leave skid marks behind on the track when the user locks up their brakes or skids.
Time Differentials - This feature will track and display the on-track time difference between any two cars.
Three NASCAR tracks present - fully modeled and texture-mapped.

Major Sub-modules
Enhance Track Creation Tool
Enhance Track Renderer for improved visuals
Lessen RAM usage of Renderer
Skid Mark Driver
Gap Time Driver

Implementation Strategy
Enhancements to the Track Creation Tool will allow tracks to be created faster and with more detail by automating some tedious tasks and compressing the track data files so more polygons can be present on a track.
Enhancements to the Track Renderer will include using a recursive method of sub-dividing polygons to counter the texture-warping and perspective-texturing problems inherent in the PlayStation hardware.
Lessening the RAM usage will involve using a smaller data type for polygons and vertices, that will expand as sections of track become visible and collapse as they move off the screen.
The Skid Mark Driver will be created using a pool of reusable skid mark polygons that will attach themselves to track sectors as they are let out due to varying conditions.
The Gap Time Driver will be created by marking specific sections of the tracks as intermediate zones that mark the time of the leader as he/she passes over it. Then as each successive car drives over that zone, the difference is taken and recorded back into the car's data structure so any two cars can compare their times (which are all relative to the leader's time).

Risks and Assumptions
Compressing anything makes the assumption that decompressing will take CPU time as well as RAM for decompression. Recursive sub-dividing may take more CPU time than a fixed subdivision function.

Dependencies
Blueprints and photo reference for NASCAR tracks

Test for Milestone Completion
To test completion, the user should be able to "drive" along a sample track with the track curves, walls, and hills drawing properly. Skid marks should appear when brakes lock or the car skids. Gap times should be displayed between any two cars.

Game Features Included
Improved texture-mapping
Skid Marks
Time Differentials

Major Product Milestones (cont)

Milestone #3 Delivery
February 28, 1997

Description
In-Car model of a NASCAR car with operating dials.
Basic Screen Interface operational (not all functions will be implemented)
Enhanced Physics
New Pause and Pit Menus
First six NASCAR tracks present

Major Sub-modules

Implementation Strategy
In-Car model will be created on Stormfront's SuperModel program.
The Screen Interface will be font-based with a static background screen, graphic pop-ups, and small side-animations.
Addition code will be added to the physics and collision models as necessary.
Pausing or Pitting will display a new pause or pit menu that will be font-based for easier foreign language translations and other modifications.
The Enhanced Physics will mainly be tuning of the physics numbers to achieve a slightly more realistic NASCAR feel without sacrificing the playability of Andretti Racing.
Currently the Physics model can be tuned using a few basic modifiable parameters:

1) Centrifugal Force Scale - this parameter increases or decreases the effect of lateral force placed on a car while cornering. (i.e. how much does a car get pushed to the outside while turning). Tuning this up can force a driver to slow more when entering turns.
2) Torque Scale - this parameter increases or decreases the amount of spin applied to a car when it's front and rear momentum are moving in opposite directions. Tuning this down will keep a car from getting out of shape too easily on ovals.
3) Base Traction/Tire Wear Rate - these parameters control how much starting traction a car's tires have and how fast they wear out and thus become more slippery. Tuning these values can make a car slip more thus forcing a driver to slow more during cornering.
4) Draft Effect - this parameter controls how much help a car receives when drafting off an opponent's car. Tuning this can either make the game more arcade-like (more draft help) or more realistic (less draft help).
5) Maximum Steering Radius - this parameter controls how far a driver can turn his steering wheel. So on longer high-speed ovals this would be turned down so a player with a digital controller can make finer adjustments while in wide sweeping turns.

Risks and Assumptions

Dependencies

To do the In-Car view well, good photographic source will be needed from EA.
Blueprints and photo reference for NASCAR tracks

Test for Milestone Completion

Should be able to drive in InCar mode with dials a moving and a slightly more realistic NASCAR racing feel. And when you pause or pit, a new menu system will greet you. The front end should be functional (press buttons, move to other screens, etc...) although not complete.

Game Features Included

Functional Dashboard.

Major Product Milestones (cont)

Milestone #4 Delivery
March 31, 1997

Description
Improved AI
Complete Menu Screen Flow
Emergency Vehicles
New Game End Transition - special camera work at the end of the race.
Driver Arm Movements - this feature will animate the driver's arms for driving and hand waving and perhaps even gear shifting.
First nine NASCAR tracks present

Major Sub-modules

Implementation Strategy
For Emergency Vehicles, a separate truck model will be created and when accidents that disable a car occur, the truck will appear at the damaged car and remove the car when no one is looking.
New Game End Transition - the race or qualifying ends with new camera work.
To animate the driver's arm, simple "motion capture" data will be kept and linked to "joints" in the arms.
Currently the AI model is very simple. A "rail" of the best driving line is prerecorded on an Mac-based version of the game. In the recording a 2-D point (world coordinates) for each sector of the track and the speed the car at that point is saved. On the Playstation, computer cars look ahead to see where they should be aiming towards next and what speed they should be going and then make the appropriate adjustments. In order to pass slower cars, a computer car may "change lanes." This simply means that it is allowed to "aim" at one of three different lines, the basic rail and one car width to the left and right of the basic rail.
Improved AI will basically be a rework of the original that will support more realistic NASCAR oval driving methods possibly using multiple driving lines that never intersect. Improving the AI will also entail a stricter adherence of the driver to the proper driving line when cornering by using additional track information to measure the distance to the inside or outside of a turn so the computer cars can maximize the driving space. In the current model the "best" driving line must always be at least one and a half car widths away from the edges of the road. This doesn't maximize the road space since the rail can't get to the very outside of turn when entering the turn and can't get to the very inside of the turn at the apex. We hope to take away this limitation allowing the rail to go right to the edges of the road and simply make the computer cars "look" and see if driving to the outside of the rail will slam them into the wall.

Risks and Assumptions
Much tuning and testing will need to be done to assure the AI works properly.

Dependencies
Blueprints and photo reference for NASCAR tracks

Test for Milestone Completion
Driver's arms should move while driving in the InCar mode. Computer cars should drive more realistically. Front-end menus should be able to be navigated although all features may not be operational. Also after big crashes, emergency vehicles should appear to take away dead cars.

Game Features Included
Active driver animations
Emergency Vehicles

Major Product Milestones (cont)

Milestone #5 Delivery
April 21, 1997

Description
Analog Control Calibration - this will allow the player to calibrate their analog steering wheel as well as their pedals, or their analog joystick.
Install new MIDI music - new MIDI music in the front end screens.
Best Line - an indicator on the track surface showing where the fastest racing line is.
Engine Blow-Ups - this feature will track when an engine expires (due to over-revving or poor workmanship) and display it with smoke and/or sound effects.
First twelve NASCAR tracks present

Major Sub-modules
Best Line editing tool

Implementation Strategy
The "Best Line" will be created on the track tool, and connecting points will mark out the path on the track. Using this string of points, either a dotted line, a rubber stained path, or something else can be drawn over the road to show the best line.
Engine Blow-Ups will simply track a player's rpm (in manual transmission mode) for too much time spent in the Red Zone and will show an explosion of smoke from the hood if the engine blows. Also periodically, a random number will be checked to see if a failure occurred. The frequency of failures could be linked to how many collisions the player has or statistically the number of failures the real team he's driving on had in the real NASCAR season. This feature may really annoy arcade players so it will be an option to be toggled.

Risks and Assumptions

Dependencies
Licensed music on which to base the composed MIDI music (by 12/31/96)
Blueprints and photo reference for NASCAR tracks

Test for Milestone Completion
To test completion, the user should be able to see a "best line". New MIDI music should play while in the front end menus. In manual transmission, you should be able to over-rev the engine and eventually blow the engine. Also, you should be able to calibrate your analog steering wheel/joystick and play with it in the game.

Game Features Included
Best Line
Engine Blow-ups

Major Product Milestones (cont)

Milestone #6 (ALPHA - NTSC) Delivery

May 27, 1997

Description

Track Telemetry - a small buffer of RAM will be set aside for lap data to be stored as the player drives over designated special sectors. The player can then go back to the front end screens to compare lap performance.
Yellow Flags/Flag Man - this will place an animating flag person at the start/finish line displaying the green, white, yellow, or checkered flag.
Video Elements Installed - EA produced video for the Championship season mode.
"Active" Loading Screen - this feature will allow animations or sounds to run while loading from the CD in the background.
All seventeen NASCAR tracks present

Major Sub-modules

Implementation Strategy

Drive a few laps and return to the telemetry screens to review your performance.
If you crash your car, the emergency vehicle should arrive and pick you up.
Video Elements will be played back through the standard Sony MDEC streaming movie player or though EA's .WVE format. Reward Animations will be modeled, rendered, and animated on 3D Studio Max and will be played back through the standard Sony MDEC streaming movie player.
Yellow Flags will be implemented in a simple format, where the computer will take hold of your car and all of the computer cars and bunch them up behind a pace car (optional depending on time and RAM). To avoid problems with pitting, while the yellow flag is out, there will be no tire wear or fuel consumption.
Flag Man - this will most likely be a simple (< 30 polygon) 3-D modeled person perched in the starter's tower, that will have motion capture data to animate his joints.

Risks and Assumptions

The motion capture data used in animating the flag person may not fit in RAM. Also time will be required to tune the motion properly.
Yellow flags will be tricky and will need lots of testing and tuning to be sure there are no ill side-effects on computer cars.

Dependencies

Video source produced by EA.
Blueprints and photo reference for NASCAR tracks

Test for Milestone Completion

Game Features Included

Track Telemetry
Yellow Flags

Major Product Milestones (cont)

Milestone #7 Delivery
June 16, 1997

Description
E3 Demo

Major Sub-modules

Implementation Strategy
For E3, we just need to clean up bugs and turn off incomplete features.

Risks and Assumptions

Dependencies
Accurate reference material of tracks.

Test for Milestone Completion
To test for completion, the player should be able to play the game without it crashing for demonstration.

Game Features Included
All seventeen NASCAR tracks

Major Product Milestones (cont)

Milestone #8 (ALPHA - PAL) Delivery

July 1, 1997

Description

Alpha version of the UK, German and French translations.

Major Sub-modules

Implementation Strategy

Conversion to PAL is pretty simple on the PlayStation. Simply calling the function SetVideoMode(PAL); sets the screens to PAL. Some timing and tuning will need to be done, but the VRAM will already be arranged so that full screen PAL will not be a problem.

As for translating all of the text to German and French, since we are making 95% of on-screen text font based, translating will only require retyping internal strings in a central header file (.h).

Risks and Assumptions

Translations may require more screen space than the English version thus requiring appropriate abbreviations from the translating company. The turnaround time may cause delays.

We hope that we may move to 25fps seemlessly but framerate doesn't improve simply because we want it to.

Dependencies

Timely delivery of accurately translated text, video, and audio.

Test for Milestone Completion

To test for completion, the player should be able to play the entire game on a PAL PlayStation in their language.

Game Features Included

Major Product Milestones (cont)

Milestone #9 (BETA) Delivery
July 28, 1997

Description
Full Game working with final art and minimal bugs.

Implementation Strategy

Risks and Assumptions
Timeliness of tuning feedback and testing is key to completing the game to this point.

Dependencies
Final Art and Video. Adequate testing support for all language versions.

Test for Milestone Completion
To test for completion, the player should be able to play the entire game with all features functioning and all final art present.

Game Features Included
All game features working.

Major Product Milestones (cont)

Milestone #10 (FINAL) Delivery
August 25, 1997

Description
Full Game working with final art and NO bugs.

Implementation Strategy
Debug like crazy!

Risks and Assumptions
Because we will be shipping all four versions (US NTSC, PAL UK, PAL French and German) simultaneously, there may be some delays in finalizing the PAL versions due to the difficulties of managing four sets of code and data (especially on a multi-disc format).

Dependencies
Final Art and Video. Sufficient testing support for all language versions.

Test for Milestone Completion
To test for completion, the player should be able to play the entire game with all features functioning and all final art present and no bugs.

Game Features Included
All game features working.

Major Product Milestones (cont)

Milestone #11 (Archive) Delivery

September 22, 1997

Description

The delivery consists of a collection of all the files and tools that are required to regenerate the Final candidate which was used for Gold. The delivery should include all source code, artwork, data files, and tools used to manipulate the files.

Implementation Strategy

The purpose of this delivery is three-fold:

(1) to establish a complete and independent system for generating the Final candidate, using Make or project files to control all variables, flags, and tools.
(2) to maintain up-to-date versions of the Development Environment checklist and Final Compilation Environment Form on this system.
(3) to copy the entire, self-referential directory structure onto removable media immediately after generation of each Final candidate.

Once the Final candidate has passed QA, the developer will complete the Project Archive Submittal Form and Project Archival Sign-off and deliver the Project Archive to EA.

Risks and Assumptions

The project must be archived after submission of the Final candidate and prior to any post-submission modifications of the source, data, or tools.

Dependencies

The Final candidate must have passed QA.

Test for Milestone Completion

An environment matching (as closely as possible) the description of the Development Environment provided in the Development Environment list will be setup. The Project Archive will be extracted onto a contiguous area of a hard disk. The steps and settings listed in the Final Compilation Environment form will be performed. The object file that is generated must match the Gold Object that was provided to Operations. Note that no assumptions can be made about software installed on the verification machine, other than Operating System and standard hardware drivers. All software tools, applications, and compilers must be provided in the project archive.

Features Included

The Game object file must be identical to the object file of the Gold.